

AppCritique Security Report

DisneyNOW

Core Report






Date Reviewed: 2019-12-01

Version Name: 4.5.2.8



Platform	Package Name	Version Code	SHA-256 Hash
Android	com.disney.datg.videoplatforms.android.watchdc	2137	73192d5a43afa93e577c5b66f059840d a24580ab3d9e1909e3321d767c35ae2b






Certificate Information

Name	Explanation
 Owner Name	Disney ABC
 Organization	Disney
 Organizational Unit	Disney ABC Television Group
 Location	Burbank, CA, US
 Validity	January 25, 2013 through June 12, 2040

Total number of flaws or potential flaws found: **15**

Functionality

Functionality Present

Name	Explanation
 Audio	This app plays audio.
 Keychain	This app stores data in the device Keystore.
 Maps	This app uses Google Maps.
 Networking	This app connects to the internet and requests the following network related permission(s): android.permission.ACCESS_NETWORK_STATE, android.permission.ACCESS_WIFI_STATE, and android.permission.INTERNET.
 Sensors	This app uses device sensors.

Functionality Not Found

Bluetooth, Camera, Calendar, Contacts, Device Administrator, Fingerprint, Google Cloud Messaging, Geofencing, Health Data, Infrared LED, In-App Purchases, Location, Microphone, NFC, Payment Services, Photos, SMS, Telephony, USB Devices

Imported Libraries

Social Networks

Facebook	×	Flickr	×	Foursquare	×
Google+	×	Instagram	×	LinkedIn	×
Pinterest	×	Tumblr	×	Twitter	×
Yelp	×				

Analytics Networks

Adjust	×	AdobeMarketingCloud	✓	AmazonAnalytics	×
AmazonInsights	×	Amplitude	×	AppBoy	×
Applause	×	Appsflyer	×	Apptimize	×
Apsalar	×	Branch	×	Countly	×
Flurry	×	GoogleAnalytics	×	Kochava	✓
Localytics	×	Mixpanel	×	MobileAppTracking	×
NewRelic	✓	Quantcast	×	Tapstream	×
Vessel	×	Webtrends	×		

Advertising Networks

AdColony	×	Adfalcon	×	Admob	✓
AmazonAds	×	Amobee	×	AppBrain	×
AppLovin	×	Appnexus	×	Axonix	×
Chartboost	×	DoubleClick	✓	FlurryAds	×

FusePowered	×	Fyber	×	IMOB	×
Inneractive	×	Kiip	×	Liquid	×
Madvertise	×	MdotM	×	mMedia	×
Mobfox	×	MobPartner	×	NativeX	×
RevMob	×	SessionM	×	Smaato	×
Tapjoy	×				

Cloud Storage

Box	×	Cloud Drive	×	Dropbox	×
GoogleDrive	×	MediaFire	×	OneDrive	×

Developer Tools

aChartEngine	×	ActiveAndroid	×	aFileChooser	×
AmazonDeviceMessaging	×	AndroidLogger	×	Annotations	×
Answers	×	Appsee	×	AsyncHttp	×
Beacon	×	BitmapCache	×	ButterKnife	×
Digits	×	Fabric	×	Firefly	✓
Parse	×	Paypal	×	PhoneGap	×
PubNub	×	Retrofit	×	Spotify	×
Stripe	×	UniversallImageLoader	×		

Permissions Requested

Permission Name	Is Used	Protection Level
android.permission.ACCESS_NETWORK_STATE		Normal
android.permission.ACCESS_WIFI_STATE		Normal
android.permission.FOREGROUND_SERVICE	*	
android.permission.INTERNET	✓	Normal
android.permission.READ_EXTERNAL_STORAGE	*	Dangerous
android.permission.RECEIVE_BOOT_COMPLETED		Normal
android.permission.WAKE_LOCK	✓	Normal
android.permission.WRITE_EXTERNAL_STORAGE	*	Dangerous

* Unable to detect use of this permission.

Security Checks

Checks Conducted

Check	Result	Explanation
Accesses External Storage	Present	<p>The app accesses the external storage directory, also referred to as the SDCard. External storage can be accessed by any app on a device with the READ/WRITE_EXTERNAL_STORAGE permission. It is therefore recommended not to store any sensitive information in external storage. External storage access is found in the following methods and classes:</p> <ul style="list-style-type: none">•com.adobe.adobepass.accessenabler.storage.FileStorageManager.<init>(Context)•com.adobe.adobepass.accessenabler.storage.StorageImporter.openDatabase(int)•com.bumptechnology.glide.load.engine.cache.ExternalCacheDiskCacheFactory\$1.getCacheDirectory()•com.disney.datg.kyln.KylnFileStorage.createFile()•com.newrelic.agent.android.AndroidAgentImpl.getEnvironmentInformation() <p>OWASP: 2016-M2-Insecure Data Storage NIAP: FDP_DAR_EXT.1.1</p>
Accesses Unique Identifiers	Present	<p>The app accesses the following unique identifiers: IMEI.</p>
Activities Accessible to Other Apps	Not Present	<p>No activities are exported, or access to all activities is restricted by use of permissions.</p> <p>OWASP: 2016-M1-Improper Platform Usage NIAP: FCS_NET_EXT.1.1</p>
Allows Data to be Backed up and Restored	Present	<p>The app allows backup of its data. A malicious actor with physical access to the device could get access to sensitive data by retrieving private files, databases, shared preferences files, caches or libraries within the app.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p>
App is Debuggable	Not Present	<p>The app is not debuggable. This protects the app against reverse engineering and the execution of arbitrary code.</p> <p>OWASP: 2016-M10-Extraneous Functionality</p>
Contains Hard-coded Cryptographic Key	Not Present	<p>No hard-coded cryptographic keys were found in the app.</p> <p>OWASP: 2016-M5-Insufficient Cryptography; 2016-M9-Reverse Engineering</p>

Contains HostnameVerifier That Accepts All Hostnames	Not Present	No weak HostnameVerifiers are found. OWASP: 2016-M3-Insecure Communication NIAP: FCS_TLSC_EXT.1.2
Contains Native Code	Present	The app loads native code libraries. Native code does not have the same security protections as Java, and is vulnerable to buffer overflows, use after free errors, and off-by-one errors. Native code can also be loaded from untrusted sources, such as a shared directory or the network. OWASP: 2016-M7-Client Code Quality NIAP: FPT_AEX_EXT.1.5
Contains Potential Hard-coded Password	Not Present	No hard-coded passwords were found in the app. OWASP: 2016-M9-Reverse Engineering
Contains Potential SQL Injection	Not Present	No potential SQL injection vulnerabilities were found. OWASP: 2016-M7-Client Code Quality
Contains Reflection Code	Present	The app contains Java reflection code. Reflection is used to instantiate new objects, invoke methods, and to get and set fields at runtime. While reflection can be used for legitimate purposes, it is also commonly employed by malware to obfuscate malicious behavior.
Contains X509TrustManager that Accepts All Certificates	Not Present	No weak X509TrustManagers are found. OWASP: 2016-M3-Insecure Communication NIAP: FIA_X509_EXT.1.1
Creates Blowfish Key with Weak Length	Not Present	The app does not create a Blowfish key with less than 128 bits in length. OWASP: 2016-M5-Insufficient Cryptography
Creates RSA Keys with Weak Modulus Length	Not Present	The app does not create an RSA key with modulus length less than 1024 bits. OWASP: 2016-M5-Insufficient Cryptography NIAP: FCS_CKM.1.1(1)
Does not Update Security Provider	Not Present	The app uses the dynamic GmsCore_OpenSSL Provider to ensure that the device's security provider is always updated. OWASP: 2016-M1-Improper Platform Usage; 2016-M5-Insufficient Cryptography

Dynamically Loads Java Classes

Present

The app dynamically loads Java classes. If these classes are loaded from untrusted sources, such as a shared directory, the network, or an app from a different developer, it could be used by an attacker to gain code execution. Dynamic loading of Java code is found in the following methods:

- androidx.appcompat.app.AppCompatActivity.onCreateViewByPrefix(Context, String, String)
- androidx.coordinatorlayout.widget.CoordinatorLayout.parseBehavior(Context, AttributeSet, String)
- androidx.fragment.app.Fragment.instantiate(Context, String, Bundle)
- androidx.fragment.app.Fragment.isSupportFragmentClass(Context, String)
- androidx.recyclerview.widget.RecyclerView.createLayoutManager(Context, String, AttributeSet, int, int)
- b.a.c.g\$b.a(String, Class, Object)
- com.adobe.mobile.MessageNotificationHandler.onReceive(Context, Intent)
- com.adobe.mobile.WearableFunctionBridge.getConfigSynchronizerClass()
- com.adobe.mobile.WearableFunctionBridge.getWearableFunctionClass()
- com.adobe.mobile.WearableFunctionBridge.isGooglePlayServicesEnabled()
- com.newrelic.agent.android.Agent.getBuildId()
- com.newrelic.agent.android.InstantApps.isInstantApp(Context)
- com.newrelic.agent.android.util.ExceptionHelper.<clinit>()
- kotlin.coroutines.jvm.internal.ModuleNameRetriever.buildCache(BaseContinuationImpl)

OWASP: 2016-M7-Client Code Quality

NIAP: FPT_TUD_EXT.1.4

Executes Environment Commands

Not Present

The app does not execute Linux-style environment commands.

OWASP: 2016-M7-Client Code Quality

The app uses a psuedo-random number generator which returns a predictable sequence of numbers that is unsuitable for security purposes. The `java.util.Random` and `java.lang.Math` classes should not be used to generate random numbers for secure use. While `SecureRandom` is the correct class for this purpose, it is recommended to avoid seeding a `SecureRandom` object. In some implementations of `SecureRandom`, seeding it may completely replace the cryptographically strong default seed. Insecure psuedo-random number generators are found in the following methods:

- `androidx.leanback.widget.nb$a.draw(Canvas, CharSequence, int, int, float, int, int, int, Paint)`
- `com.disney.datg.novacorps.player.ad.AbcClientSideAds.prepare(MediaPlayer, Walkman, Walkman, SurfaceHolder, WebView, List, String, AdEvent)`
- `com.disney.datg.novacorps.player.ad.AbcServerSideAds.prepare(MediaPlayer, Walkman, WebView, List, String)`
- `com.disney.datg.novacorps.player.ad.extension.AdExtensionsKt.formatAdErrorUrl(Ad, VastErrorCode, int)`
- `com.moat.analytics.mobile.abc.a.<init>(Context, int)`
- `com.moat.analytics.mobile.abc.l.(Exception)`
- `com.newrelic.agent.android.analytics.EventManagerImpl.addEvent(AnalyticsEvent)`
- `com.newrelic.agent.android.connectivity.CatFacade.randomNumber()`
- `com.newrelic.agent.android.tracing.Trace.<init>()`
- `com.newrelic.agent.android.tracing.Trace.<init>(String, UUID, TraceMachine)`
- `com.nielsen.app.sdk.P.l()`
- `com.nielsen.app.sdk.P.m()`
- `kotlin.random.AbstractPlatformRandom.nextBits(int)`
- `kotlin.random.AbstractPlatformRandom.nextBoolean()`
- `kotlin.random.AbstractPlatformRandom.nextBytes(byte)`
- `kotlin.random.AbstractPlatformRandom.nextDouble()`
- `kotlin.random.AbstractPlatformRandom.nextFloat()`
- `kotlin.random.AbstractPlatformRandom.nextInt()`
- `kotlin.random.AbstractPlatformRandom.nextInt(int)`
- `kotlin.random.AbstractPlatformRandom.nextLong()`
- `okhttp3.internal.ws.RealWebSocket.<init>(Request, WebSocketListener, Random, long)`
- `okhttp3.internal.ws.WebSocketWriter.writeControlFrame(int, ByteString)`
- `okhttp3.internal.ws.WebSocketWriter.writeMessageFrame(int, long, boolean, boolean)`

OWASP: 2016-M5-Insufficient Cryptography

NIAP: FCS_RBG_EXT.1.1

Logs Information	Present	<p>The app prints logging information to the system log. While apps often log information for debugging purposes, this should generally be removed before an app is put into production. No sensitive information, such as keys or authentication tokens, should ever be written to the system log.</p> <p>OWASP: 2016-M2-Insecure Data Storage</p> <p>NIAP: FCS_CFG_EXT.1.2</p>
Providers Accessible to Other Apps	Not Present	<p>The app does not contain content providers, no content provider is exported, or access to all content providers is restricted by use of permissions.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p> <p>NIAP: FMT_CFG_EXT.1.2</p>
Receivers Accessible to Other Apps	Not Present	<p>The app does not contain receivers, no receivers are exported, or access to all exported receivers is restricted by use of permissions.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p>
Requests Root Access	Not Present	<p>The app does not request root access.</p> <p>OWASP: 2016-M8-Code Tampering</p>
Services Accessible to Other Apps	Not Present	<p>The app does not contain services, no services are exported, or access to all services is restricted by use of permissions.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p>
SMS CVE-2014-8610	Not Present	<p>The app does not send text messages or has the required SMS permission. It is protected from vulnerability CVE-2014-8610, where an unprivileged app can resend all the SMS stored in the user's phone to their corresponding recipients or senders without user interaction.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p>
Source Code is not Obfuscated	Present	<p>The app does not obfuscate the majority of its code by renaming classes, fields, methods, and variables. This allows an adversary or competitor to decompile the app into near-original source code. It is recommended to obfuscate the app's code using a tool such as ProGuard to make it more difficult to reverse engineer.</p> <p>OWASP: 2016-M9-Reverse Engineering</p>
Uses Cipher That Does not Provide Integrity	Not Present	<p>This app does not use a cipher that does not provide data integrity.</p> <p>OWASP: 2016-M5-Insufficient Cryptography</p>
Uses Dangerous Permissions	Not Present	<p>No dangerous permissions were requested by the app.</p> <p>OWASP: 2016-M1-Improper Platform Usage</p>

Uses DES or 3DES Cipher	Not Present	The app does not use the DES or 3DES cipher. OWASP: 2016-M5-Insufficient Cryptography
Uses Electronic Code Book Mode Cipher Mode	Not Present	This app does not use ciphers with the Electronic Code Book (ECB) mode. OWASP: 2016-M5-Insufficient Cryptography
Uses MD5 Hashing Algorithm	Present	The app uses the weak MD5 hashing algorithm. The MD5 algorithm is dangerous if used for sensitive data because it is highly vulnerable to collision attacks. OWASP: 2016-M5-Insufficient Cryptography
Uses NullCipher	Not Present	This app does not use NullCipher. OWASP: 2016-M5-Insufficient Cryptography
Uses Object Deserialization	Present	<p>The app calls the <code>java.io.ObjectInputStream.readObject()</code> method to deserialize objects into memory. Object deserialization is a common source of vulnerabilities, particularly when the object may be from an untrusted source. It is recommended to avoid object deserialization when possible, or otherwise to harden the <code>ObjectInputStream</code> against attacks. One strong hardening technique is to override the <code>resolveClass()</code> to only allow expected classes. This can be implemented by subclassing <code>ObjectInputStream</code>. Object deserialization is found in the following methods:</p> <ul style="list-style-type: none"> •<code>androidx.work.Data.a(byte)</code> •<code>com.adobe.adobepass.accessenabler.models.Mvpd.deserialize(String)</code> •<code>com.adobe.adobepass.accessenabler.storage.FileStorageManager.getClientInfo(String)</code> •<code>com.adobe.adobepass.accessenabler.storage.FileStorageManager.readFromStorage()</code> •<code>com.comscore.android.vce.r.e(r\$a)</code> <p>OWASP: 2016-M7-Client Code Quality</p>
Uses RSA Encryption Algorithm Without Padding	Not Present	The app does not use the RSA algorithm without padding. OWASP: 2016-M5-Insufficient Cryptography
Uses SHA1 Hashing Algorithm	Present	The app uses the weak SHA1 hashing algorithm. The SHA1 algorithm is dangerous if used for sensitive data because it is vulnerable to collision attacks. OWASP: 2016-M5-Insufficient Cryptography

Weak Construction of Socket Factory

Present

The app creates a SSL socket factory that may be vulnerable to man-in-the-middle (MitM) attacks. The `SSLConnectionFactory` class creates sockets that do not automatically perform certificate hostname validation, leaving the burden on the developer to manually create and use a `HostnameVerifier`. Failure to do so could allow an attacker to MitM the SSL socket connection by presenting legitimate signed certificate for a different hostname. The `SSLCertificateConnectionFactory` class creates sockets that automatically perform hostname validation, but only when instantiated with 'String host'. `SSLCertificateConnectionFactory` also has a method 'getInsecure' that returns a SSL socket factory with all security checks disabled, which would allow an attacker to perform a MitM attack with any certificate. Google recommends choosing the highest level networking API possible, such as `HttpsURLConnection`, because the higher level APIs perform these security checks automatically. The following methods create SSL socket factories that either do not automatically perform hostname validation or do not automatically perform any security validation:

- `com.disney.datg.rocket.TlsSocketFactory.createSocket()`
- `com.disney.datg.rocket.TlsSocketFactory.createSocket(InetAddress, int)`
- `com.disney.datg.rocket.TlsSocketFactory.createSocket(InetAddress, int, InetAddress, int)`
- `com.disney.datg.rocket.TlsSocketFactory.createSocket(Socket, String, int, boolean)`
- `com.disney.datg.rocket.TlsSocketFactory.createSocket(String, int)`
- `com.disney.datg.rocket.TlsSocketFactory.createSocket(String, int, InetAddress, int)`
- `com.google.android.gms.internal.ads.zzbex.createSocket(InetAddress, int)`
- `com.google.android.gms.internal.ads.zzbex.createSocket(InetAddress, int, InetAddress, int)`
- `com.google.android.gms.internal.ads.zzbex.createSocket(Socket, String, int, boolean)`
- `com.google.android.gms.internal.ads.zzbex.createSocket(String, int)`
- `com.google.android.gms.internal.ads.zzbex.createSocket(String, int, InetAddress, int)`
- `okhttp3.internal.connection.RealConnection.connectTls(ConnectionSpecSelector)`

OWASP: 2016-M3-Insecure Communication

Weak RSA Modulus Length of App Signing Certificate

Not Present

The app is signed with a key of 2048 bit length or greater, as recommended by Google. The private signing key is protected.

OWASP: 2016-M5-Insufficient Cryptography

Weakly Configured XML Parser

Present

The app may use weakly configured XML parsing libraries. If XML is parsed from an untrusted source, this can lead to XML External Entity (XXE) and Denial of Service (DoS) attacks. It is recommended to use the secure processing feature to prevent DoS attacks, and disallow document type declaration (DTD) to protect against most XXE attacks. If it is not possible to disallow DTD, then disable external entities and external doctypes. Weakly configured XML parsing is found in the following methods:

- com.adobe.adobepass.accessenabler.models.AuthenticationToken.parseXmlHelper(String)
- com.adobe.adobepass.accessenabler.models.AuthorizationToken.parse(String)
- com.adobe.adobepass.accessenabler.models.AuthorizationToken.preParse(String)
- com.adobe.adobepass.accessenabler.models.PreauthorizationCache.parsePreauthorizeMessage(String)
- com.adobe.adobepass.accessenabler.models.Requestor.parseConfigFromXml(String, String)
- com.disney.datg.novacorps.auth.AccessEnablerManager.translateToken(String)

OWASP: 2016-M7-Client Code Quality

Webview Contains JavaScript Interface

Present

The app exposes an interface to access internal Java methods from JavaScript in a WebView. This interface should be used with extreme caution. If the WebView displays untrusted content, an attacker could remotely execute these methods.

OWASP: 2016-M7-Client Code Quality

Hard-coded Values Found

URLs	Country
http://abc.go.com/watchabc-overview?webview=true	US
http://b.scorecardresearch.com/p2	US
http://b.scorecardresearch.com/rs/mobile/ntv/vce_st.js	US
http://b.voicefive.com/rs/sdk/gg.js	US
http://content.uplynk.com/player/assetinfo/	US
http://crl.verisign.com/pca3.crl0)U	US
http://logo.verisign.com/vslogo.gif0U%0++04+(0&0\$+0⚡http://ocsp.verisign.com01U*0(0&⚡\$⚡	*
http://ns.adobe.com/creatorAtom/1.0/	*
http://ns.adobe.com/xap/1.0/	*
http://ns.adobe.com/xap/1.0/mm/	*
http://ns.adobe.com/xap/1.0/sType/ResourceEvent#	*
http://ns.adobe.com/xmp/1.0/DynamicMedia/	*
http://priv-policy.imrworldwide.com/priv/amazon/us/en/optout.html	US
http://priv-policy.imrworldwide.com/priv/amazon/us/en/optout_legacy.html	US
http://priv-policy.imrworldwide.com/priv/mobile/us/en/optout.html	US
http://purl.org/dc/elements/1.1/	US
http://schemas.android.com/aapt	*
http://schemas.android.com/apk/res-auto	*
http://schemas.android.com/apk/res/android	*
http://schemas.microsoft.com/DRM/2007/03/protocols/AcquireLicense	US
http://search.yahoo.com/mrss/	US
http://search.yahoo.com/mrss/	US
http://segment-data-us-east.zqtk.net/%s?url=%s&c2=%s&c12=%s&ns_ap_bi=%s&ns_ap_sv=%s&ns_ap_an=%s&ns_ap_ver=%s&ns_ap_pn=android	US
http://streams.adobeprimetime.com/v2	US
http://udm.scorecardresearch.com/offline	US
http://www.example.com	US
http://www.google.com	US
http://www.linotype.com/license	US
http://www.nielsen.com/digitalprivacy.	US
http://www.w3.org/1999/02/22-rdf-syntax-ns#	US
http://www.w3.org/ns/ttml#parameter	US

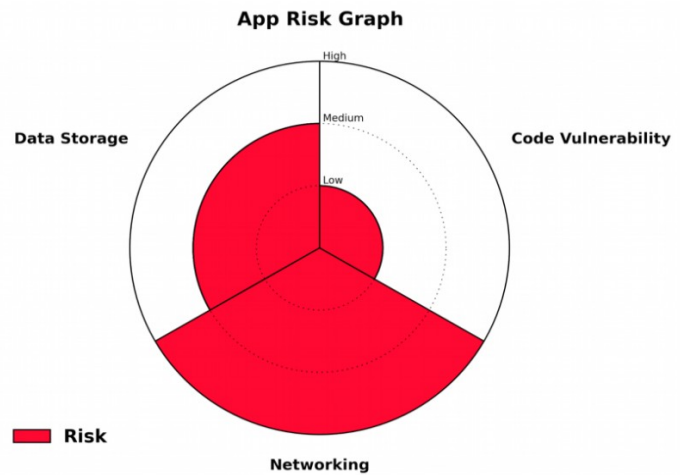
https://api.auth.adobe.com	US
https://assets.adobedtm.com/b213090c5204bf94318f4ef0539a38b487d10368/scripts/satellite-55f19fe131336300140012cb.json	US
https://assets.adobedtm.com/b213090c5204bf94318f4ef0539a38b487d10368/scripts/satellite-55f19fe33632320017000d7d.json	US
https://content.uplynk.com/wv	US
https://csi.gstatic.com/csi	US
https://github.com/ReactiveX/RxJava/wiki/Plugins	US
https://github.com/ReactiveX/RxJava/wiki/What	US
https://github.com/ReactiveX/RxJava/wiki/What\	US
https://goo.gl/J1sWQy	US
https://google.github.io/ExoPlayer/faqs.html#what-do-player-is-accessed-on-the-wrong-thread-warnings-mean	*
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/mraid/v2/mraid_app_banner.js	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/mraid/v2/mraid_app_expanded_banner.js	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/mraid/v2/mraid_app_interstitial.js	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/native_ads.html	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/production/native_ads.js	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/production/sdk-core-v40-impl.html	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/production/sdk-core-v40-impl.js	US
https://googleads.g.doubleclick.net/mads/static/mad/sdk/native/sdk-core-v40-loader.html	US
https://hal.testandtarget.omniture.com	US
https://imasdk.googleapis.com/admob/sdkloader/native_video.html	US
https://pagead2.googlesyndication.com/pagead/gen_204?id=gmob-apps	US
https://px.moatads.com/pixel.gif?e=0&i=MOATSDK1&ac=1	US
https://sb.scorecardresearch.com/p2	US
https://sb.scorecardresearch.com/rs/mobile/ntv/vce_st.js	US
https://sb.voicefive.com/rs/sdk/b.html	US
https://sb.voicefive.com/rs/sdk/gg.js	US
https://support.google.com/dfp_premium/answer/7160685#push	US
https://udm.scorecardresearch.com/offline	US
https://www.edgedatg.com	US
https://www.go.com	US
https://www.google.com/dfp/debugSignals	US
https://www.google.com/dfp/inAppPreview	US

https://www.google.com/dfp/linkDevice	US
https://www.google.com/dfp/sendDebugData	US
https://www.googleapis.com/auth/appstate	US
https://www.googleapis.com/auth/datastoremobile	US
https://www.googleapis.com/auth/drive	US
https://www.googleapis.com/auth/drive.appdata	US
https://www.googleapis.com/auth/drive.apps	US
https://www.googleapis.com/auth/drive.file	US
https://www.googleapis.com/auth/fitness.activity.read	US
https://www.googleapis.com/auth/fitness.activity.write	US
https://www.googleapis.com/auth/fitness.blood_glucose.read	US
https://www.googleapis.com/auth/fitness.blood_glucose.write	US
https://www.googleapis.com/auth/fitness.blood_pressure.read	US
https://www.googleapis.com/auth/fitness.blood_pressure.write	US
https://www.googleapis.com/auth/fitness.body.read	US
https://www.googleapis.com/auth/fitness.body.write	US
https://www.googleapis.com/auth/fitness.body_temperature.read	US
https://www.googleapis.com/auth/fitness.body_temperature.write	US
https://www.googleapis.com/auth/fitness.location.read	US
https://www.googleapis.com/auth/fitness.location.write	US
https://www.googleapis.com/auth/fitness.nutrition.read	US
https://www.googleapis.com/auth/fitness.nutrition.write	US
https://www.googleapis.com/auth/fitness.oxygen_saturation.read	US
https://www.googleapis.com/auth/fitness.oxygen_saturation.write	US
https://www.googleapis.com/auth/fitness.reproductive_health.read	US
https://www.googleapis.com/auth/fitness.reproductive_health.write	US
https://www.googleapis.com/auth/games	US
https://www.googleapis.com/auth/games.firstparty	US
https://www.googleapis.com/auth/games_lite	US
https://www.googleapis.com/auth/plus.login	US
https://www.googleapis.com/auth/plus.me	US
https://z.moatads.com/	US

Upgrade to App Vulnerability Assessment

Thank you for trying the free AppCritique scan! Data breaches lead to loss of intellectual property, litigation, customer dissatisfaction and loss in revenue. Many significant vulnerabilities cannot be detected by our Free Report. AppCritique offers the **App Vulnerability Assessment (AVA)** as a deep dive of your app conducted by the AppCritique experts. The AVA service includes:

- Expert analysis of your app with risk assessment write-ups
- Recommendations and remediations
- Q&A session between your app developers or assurance team and the expert AppCritique analysts
- AVA checks include:
 - **Code Vulnerabilities**
 - Inter-app communications
 - Components vulnerable to manipulation
 - Unauthenticated or unfiltered input
 - Local SQL injection
 - Unsafe native code
 - Dynamically loaded code
 - Hard-coded credentials
 - Deprecated cryptography
 - **Data Storage**
 - Data accessible in unencrypted backups
 - Publicly accessible sensitive information
 - Credentials outside secure store
 - Data privacy analysis
 - Side channel data leakage
 - **Networking**
 - Certificate validation issues
 - Unencrypted protocols
 - Weak endpoint encryption
 - Back-end analysis
 - Man-in-the-middle attack scenarios



Sample App Risk Graph

Contact Us

Email us at AppCritique@bah.com to set up your App Vulnerability Assessment!